University of Nevada, Las Vegas Computer Science 456/656 Fall 2023

Answers to Third Examination November 22, 2023

1. True or False. T = true, F = false, and O = open, meaning that the answer is not known science at this time.

   (i) **T** The context-free grammar equivalence problem is co-$\mathcal{RE}$.

   (ii) **T** Let $L = \{(G_1, G_2)\}$: G_1 and G_2 are not equivalent. Then L is recursively enumerable.

   (iii) **T** The factoring problem for unary numerals is $\mathcal{P}$–TIME

   (iv) **T** The set of all binary numerals for prime numbers is in $\mathcal{P}$–TIME.

   (v) **F** If L is a recursively enumerable language, there must be a machine which enumerates L in canonical order.

   (vi) **F** The set of all positive real numbers is countable.

   (vii) **T** Let L be a recursive language over an alphabet Σ, and M a machine that decides L. For any n, let $F(n)$ be the maximum number of steps M needs to decide whether a given string in Σ^* of length n is in L. Then F must be recursive.

   (viii) **F** Let L be a recursively enumerable language over an alphabet Σ, and M a machine that accepts L. For any n, let $G(n)$ be the maximum number of steps M needs to accept any string in L of length n. Then G must be recursive.

   (ix) **T** For any alphabet Σ, the set of all recursively enumerable languages over Σ is countable.

   (x) **T** If L is a context-free language over the unary alphabet, then L must be regular.

   (xi) **F** The union of any two undecidable languages is undecidable.

   (xii) **T** co-$\mathcal{P}$

2. Give a definition of a recursive real number. (There is more than one correct definition.)

3. (a) $x \in \mathbb{R}$ is recursive means that there is a machine that writes the decimal expansion of x.

   (b) $x \in \mathbb{R}$ is recursive means that the function D, where $D(n)$ is the D^{th} digit of the decimal expansion of x, is recursive.

   (c) $x \in \mathbb{R}$ is recursive means that, for any fracton y, the question of whether $x < y$ is decidable.

   Which of these languages (problems) are **known** to be $\mathcal{NP}$-complete? If a language, or problem, is known to be $\mathcal{NP}$-complete, fill in the first circle. If it is either known not be be $\mathcal{NP}$-complete, or if whether it is $\mathcal{NP}$-complete is not known at this time, fill in the second circle.

$\otimes$ $\bigcirc$ Boolean satisfiability.
$\bigcirc$ $\otimes$ 2SAT.
$\otimes$ $\bigcirc$ 3SAT.
$\otimes$ $\bigcirc$ Subset sum problem.
$\bigcirc$ $\otimes$ Generalized checkers, i.e. on a board of arbitrary size.

$\otimes$ $\bigcirc$ Traveling salesman problem.
$\bigcirc$ $\otimes$ Rush Hour: https://www.youtube.com/watch?v=HI0rlp7tiZ0
$\otimes$ $\bigcirc$ Dominating set problem.
$\bigcirc$ $\otimes$ Strong connectivity of directed graphs.
$\bigcirc$ $\otimes$ Circuit value problem, CVP.
$\bigcirc$ $\otimes$ C++ program equivalence.
$\otimes$ $\bigcirc$ Partition.
$\bigcirc$ $\otimes$ Regular language membership problem.
$\otimes$ $\bigcirc$ Block sorting.

4. State the pumping lemma for regular languages.

For any regular language L,
there exists an integer p,‘
such that for any $w \in L$ of length at least p,
there exist strings x, y, z such that the following statements hold:
   1. $xyz = w$
   2. $|xy| \leq p$
   3. $|y| > 0$
   4. for any integer $i \geq 0$, $xy^iz \in L$.

5. Give a polynomial time reduction of 3SAT to the independent set problem.

Let $E = C_1 * C_2 * \cdots C_k$ be Boolean expression in 3-CNF form For any i, let $C_i = t_{i,1} + t_{i,2} + t_{i,3}$ where each $t_{i,p}$ is either a variable or the negation of a variagle. Let G be the graph with $3k$ vertices $\{v_{i,j}\}$ each labeled with one term of E. Let there be an edge from $v_{i,p}$ to $v_{j,q}$ if either $i = j$ or $t_{i,p} * t_{j,q}$ is a contradiction. Then E is satisfiable if and only if G has an independent set of order k.

6. Prove that any recursively enumerable language is accepted by some machine.

Let $w_1, w_2, \ldots$ be a recursive enumeration of L. The following program acceps L.

read w
for i from 1 to ∞
    if($w = w_i$)
        write 1 and halt

7. Prove that any recursive language can be enumerated in canonical order by some machine.

Let $L \subseteq \Sigma^*$ be a recursive language. Let $w_1, w_2 \ldots$ be a canonical order enumeration of Σ^* in canonical order. The following program enumerates L in canonical order.

for all i from 1 to ∞

if($w_i \in L$)

write w_i

8. Consider G, the following context-free gammar with start symbol E. Stack states are indicated.

1. $E \to E_{1,11} +_2 E_3$
2. $E \to E_{1,11} -_4 E_5$
3. $E \to E_{1,3,5,11} *_6 E_7$
4. $E \to -_8 E_9$
5. $E \to (_{10} E_{11})_{12}$
6. $E \to x_{13}$

(a) Below are the tables of an LALR parser for G. Fill in the missing colums.

	x	$+$	$-$	$*$	(	)	\$	E
0	s13				s10			1
1				s6			**halt**	
2	s13				s10			3
3				s6		r1	r1	
4	s13				s10			5
5				s6		r2	r2	
6	s13				s10			7
7				r3		r3	r3	
8	s13				s10			9
9				r4		r4	r4	
10	s13				s10			11
11				s6		s12		
12				r5		r5	r5	
13				r6		r6	r6	

(b) Give a complete computation of the parser if the input string is $x - x * -(-x + x)$.

$\$_0$	$x - x * -(-x + x)\$$		
$\$_0 x_{13}$	$-x * -(-x + x)\$$	s_{13}	
$\$_0 E_1$	$-x * -(-x + x)\$$	$r6$	6
$\$_0 E_1 -_4$	$x * -(-x + x)\$$	$s4$	6
$\$_0 E_1 -_4 x_{13}$	$* - (-x + x)\$$	$s13$	6
$\$_0 E_1 -_4 E_5$	$* - (-x + x)\$$	$r6$	66
$\$_0 E_1 -_4 E_5 *_6$	$-(-x + x)\$$	$s6$	66
$\$_0 E_1 -_4 E_5 *_6 -_8$	$(-x + x)\$$	$s8$	66
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10}$	$-x + x)\$$	$s10$	66
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} -_8$	$x + x)\$$	$s8$	66
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} -_8 x_{13}$	$+x)\$$	$s13$	66
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} -_8 E_9$	$+x)\$$	$r6$	666
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11}$	$+x)\$$	$r4$	6664
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11} +_2$	$x)\$$	$s2$	6664
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11} +_2 x_{13}$	$)\$$	$s13$	6664
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11} +_2 E_3$	$)\$$	$r6$	66646
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11}$	$)\$$	$r1$	666461
$\$_0 E_1 -_4 E_5 *_6 -_8 (_{10} E_{11})_{12}$	$\$$	$s12$	666461
$\$_0 E_1 -_4 E_5 *_6 -_8 E_9$	$\$$	$r5$	6664615
$\$_0 E_1 -_4 E_5 *_6 E_7$	$\$$	$r4$	66646154
$\$_0 E_1 -_4 E_5$	$\$$	$r3$	666461543
$\$_0 E_1$	$\$$	$r2$	6664615432

HALT 6664615432

9. Fill in the following table, showing which operations are closed for each class of languages. In each box, write **T** if it is known that that language class is closed under that operation, **F** it is known that that class is not closed under that operation, and **O** if neither of those is known.

language class	union	intersection	concatenation	Kleene closure	complementation
$\mathcal{NC}$					
context-free					
$\mathcal{NP}$					
recursive					
co-$\mathcal{RE}$					
undecidable					

10. Consider the following well-known complexity classes:

$\mathcal{NC} \subseteq \mathcal{P}$–TIME$\subseteq \mathcal{NP} \subseteq \mathcal{P}$–SPACE $\subseteq$**EXP**–TIME $\subseteq$**EXP**–SPACE

(a) Which of the above complexity classes is the smallest class which is known to contain SAT, the Boolean satisfiability problem?

$\mathcal{NP}$

(b) Which of the above complexity classes is the smallest class which is known to contain the connectivity problem for graphs?

$\mathcal{P}$–TIME

(c) Which of the above complexity classes is the smallest class which is known to contain the context-free language membership problem?

$\mathcal{NC}$

(d) Which of the above complexity classes is the smallest class which is known to contain every sliding block problem?

$\mathcal{P}$–SPACE

(e) Which of the above complexity classes is the smallest class which is known to contain integer matrix multiplication?

$\mathcal{NC}$

(f) We say that a computer program is straight-line if no portion of the code can be executed more than once. That implies that the code contains no loops or recursion, and no GOTO from one line of the code to an earlier line. Which of the above complexity classes is the smallest class which contains the problem of determining whether the output of a straight-line program is zero?

$\mathcal{P}$–TIME